EEL4924 Design II

Final Design Report

Portable Vitals

Shot Through the Heart

Submitted By:

Madison Breese and Raymond Salzmann

Table of Contents

Table of Figures and Tables

Background Information

An electrocardiogram (ECG) is used to measure a heart's electrical activity. From an ECG you can gain information about a person's heart, such as heart rate and heart rhythms. A typical ECG machine is large, around the size of a laptop, and requires 12 electrodes to be attached to a person's chest and various limbs. This is inconvenient, and not feasible for people with heart conditions that may need to be monitoring their heart's activity throughout the day. A portable ECG monitor is simple, convenient, and much more economical. Most products on the market today require only one to two electrodes and are approximately the size of a cellular phone. This allows users to monitor their heart activity while at home, work, etc.

Another important health metric is body temperature, as it is used to help diagnose a number of diseases and illnesses and allows doctors to analyze the effectiveness of treatments. A fever is the most common reason for an increase in body temperature. The body changes its normal temperature to support the body's own defense mechanisms. For adults a normal body temperature is between the range of 97.7 – 99.5 °F; a fever begins at 100.9 °F and hypothermia is any temperature less than 95 °F.

Body mass index (BMI) is an attempt to quantify the amount of tissue mass (muscle, fat, and bone) in an individual. It is frequently used to determine how much an individual's body weight differences for what is normal/desirable for a person's height. BMI is derived from an individual's height and weight and is split into three categories: underweight, normal (healthy), and overweight. The formula used to calculate BMI is $\frac{mass\ (lbs.)}{height\ (in.)^2}\ x\ 703\ [\frac{lbs.}{in.^2}]$. Commonly accepted BMI ranges are underweight: under 18.5 lbs./in.^{2}, normal weight: 18.5 – 25 lbs./in.^{2}, and overweight: over 25 lbs./in.^{2}.

Project Definition

The team will design and build a portable vitals monitor that allows individual's to track their heartrate, body temperature, and body mass index. The product will consist of a handheld device with a graphic display to display the user's vitals, a keypad for the user to select from preprogrammed menu items and input necessary information, and three electrodes to gather heart data. A green LED will indicate when power is being delivered to the monitor; power will come from a 3.7 V battery. The graphic display will display one of the health metric's depending on the user's input on the menu screen:

- A: ECG graph and heartrate
- B: Thermometer
- C: BMI calculator
- D: Return to menu

The monitor will be controlled by a microcontroller and will utilize the microcontroller's analog-to-digital (ADC) and serial peripheral interface (SPI) modules.

Major Objectives

I. Construct a device that is easily portable and simple to use
II. Develop the ~1 mV signal collected from the electrodes into an analog signal that can be properly analyzed by the MCU
    i. Interpret the electrode data in order to do the necessary calculations to obtain user's current heartrate
    ii. Display the ECG graph and user's current heartrate on graphic OLED display
III. Design a reliable way to measure body temperature

Technical Objectives

Hardware

I. Amplification Circuitry:
    i. Differential amplifier to combine and amplify signal from electrodes
II. Filtering Circuitry:
    i. Low pass filter to filter out the 50-60 Hz of external noise from electrical components
    ii. High pass filter to filter out respiratory noise from user
III. Thermometer Circuitry: to accurately measure body temperature
IV. Power Circuitry: to supply necessary voltage to all circuits
V. Battery Recharge Circuitry: to recharge 3.7 V battery used to power device

Software

I. Displaying EKG Waveform
    i. Utilizes MCU's ADC and SPI modules to collect data from amplification circuitry and display on screen
II. Calculating Heartrate
    i. Utilizes MCU's ADC module to find peaks in data that signify a heartbeat
III. Calculating Body Temperature
    i. Utilizes MCU's ADC module to collect data from thermometer circuitry
IV. Calculating BMI
    i. Utilizes MCU's GPIO ports to receive user input
V. OLED Graphic Display
    i. Utilizes MCU's SPI module
VI. Keypad
    i. Utilizes MCU's GPIO ports to determine which button was pressed
VII. Interfacing MCU with different buttons and switches used throughout the system

Product Expectations

I. Graphic display that will allow users to choose which health metric to view
II. ECG graph and heartrate
    i. Audio sound when heartbeat detected
III. Body temperature
    i. Informs user if body temperature is in normal, fever, or hypothermic range

IV. BMI Calculator
    i. Informs user if BMI is in normal, overweight, or underweight range

Analysis of Competitive Products

Currently, there are several vital signs monitors on the market. However, the majority of these monitors are not super portable, and they are costly. The average price of these monitors is ~$300.

Concept Selection

Product Selection

Microprocessor:	TI MSP430FG439	TI MSP430F6436	TI MSP430FR2355
Clock Speed [MHz]	8	8	24
RAM [KB]	60	18	32
Flash [kB]	2	128	4
GPIO	48	74	44
ADC	12-bit SAR	12-bit SAR	12-bit SAR
DAC	2 12-bit	2 12-bit	4 12-bit
LCD Drivers	yes	yes	no
Timers	2 16-bit	4 16-bit	4 16-bit
SPI	1	4	4
Real-Time Clock	no	yes	yes
Cost [$]	6.24	4.92	4.53

Table 1 Microcontroller Selection

The processor chosen for the project is the MSP430FR2355. It had all of the specs required for the scope of our project. When originally researching microcontrollers, it was unknown if an LCD module was going to be needed for the display. After further research of displays it was found that both options had a built-in controller, thus eliminating the need for the microcontroller to also contain an LCD module. The MSP430FR2355 also has a much faster clock speed than the other options.

Display:	NHD Graphic LCD	NHD Graphic OLED
Part #	NHD-240128WG-ATMI-VZ#	NHD-3.12-25664UCB2
Display Type	STN-Super Twisted Nematic	OLED-Passive Matrix
Diagonal Screen Size [in.]		3.12
Pixels	240 x 128	256 x 64
Interface	Parallel, 8-bit	Parallel, 8-bit, SPI
Power Supply [V]	5	3.3
Built-In Controller	yes	yes
Cost [$]	61.42	34.65

Table 2 Display Selection

Both display options met the requirements to display the user's heart rate and EKG graph. However, the graphic OLED requires only 3.3V of power, while the graphic LCD requires 5V of power. The OLED is also approximately half the price of the graphic LCD.

Hardware Concept Design

I. Power Supply: The power supply design will provide the necessary power requirements of the portable vitals monitor. The design requirements are:
   - +/- 10 V:
     - to obtain heart data from electrodes
   - 3.3 V:
     - Thermometer circuitry
     - Microcontroller
   - 5 V:
     - Display Screen
     - Power Indicating LED

II. Graphic Display: The OLED display will be graphic in order to display the ECG graph to the user

III. Keypad: The keypad input will be a standard 4x4 matrix keypad

Software Concept Design

I. Microprocessor: The microprocessor will be selected to meet all of the specs required for the scope of the project

Project Architecture

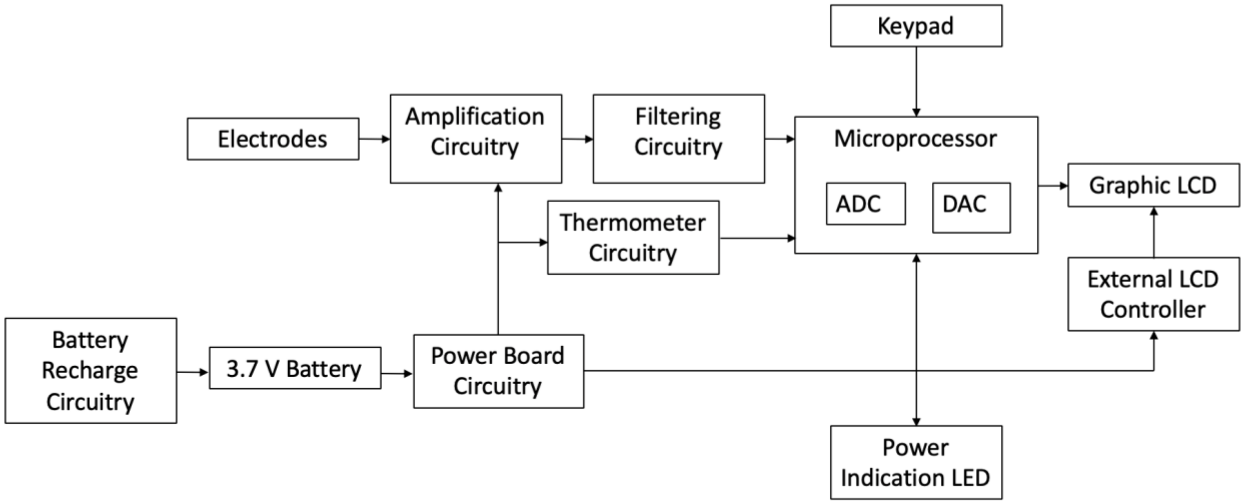

Figure 1 Hardware Block Diagram

Hardware

Battery Recharge Circuitry

The purpose of the battery recharge circuit is to recharge the 3.7 V battery. When the battery's charge is at ~3.8 V, the circuit will begin to charge the battery. When the battery's charge reaches ~4.18 V, the circuit will stop charging the battery.

Power Board Circuitry:

The purpose of the power board is to be a hub that supplies enough current and the necessary voltage to all of the other necessary boards. The main component of the power board is the TPS65130 split-rail converter which takes the +3.7V input from our Lithium Ion battery and converts it to +10V and -10.5V. These output voltages are adjustable based on the values of a select few supplementary component, however we decided that the output voltages stated before were satisfactory for what we needed. The main reason we used this part was the need to have a negative voltage for the rails of the TL071 Operational Amplifiers used in the Amplifier & Filters board. Once the desired output voltages are achieved using the TPS65130, we can regulate the +10V down to +5V using an LM7805 voltage regulator and the +5V down to +3.3V using an LM3940 voltage regulator. We are also able to regulate the -10V down to -5V using a 7905C voltage regulator. The combination of all of these voltages enables us to provide the necessary power to all of our supplementary boards from one central unit.

Amplification and Filtering Circuitry:

The purpose the amplifiers and filters are to take the raw electrical signal receiver from the three electrodes attached to the user's body and output the ECG data and its graph in a such a way that is useful to learn about the user's cardiovascular health. This is accomplished through the use of multiple filters and amplifiers. First, a differential amplifier combines the electrode signals to make it one signal. This signal is on the order of ~1mV Vpp and contains noise from several sources, such as the user's muscles, respiration and the 60Hz power used in the United States. To eliminate all noise, the signal is sent through a low-pass filter with a cut-off frequency of roughly 30 Hz. The signal is then sent though a high-pass filter to eliminate a phenomenon known as "baseline wander". Baseline wander causes the ECG signal to oscillate in such a way that it mimics a since wave at approximately 0.5 Hz. This was eliminated by using a high-pass filter at roughly 1 Hz. The signal is then sent through a non-inverting amplifier to allow the signal to be much more apparent on the tool being used to measure the signal. In our case, we are reading this signal from the ADC module on a TI MSP430FR2355 microcontroller, so the desired V_{PP} is roughly 3.3V. Lastly, the signal is fed through an operational amplifier that adds a DC bias so the signal fits nicely in the 0 V – 3.3 V range required by the ADC channel.

Thermometer Circuitry:

To obtain user's body temperature, a TMP36 temperature sensor is used. The output of the sensor is sent through circuitry that converts that converts the voltage output of the sensor from 750 mV at 25°C to 1 mV/°F. The transfer equation for the circuit is given by: $V_{out} = \left(\frac{R_{11}+R_{12}}{R_7+R_8+R_9+R_{10}+R_{11}+R_{12}}\right)\left(1+\frac{R_5}{R_2+R_3+R_4}\right)(TMP36) - (\frac{R_5}{R_2+R_3+R_4})(\frac{V_S}{2})$

Keypad:

The purpose of the keypad is to obtain user input for the menu, to determine which health metric the user wishes to observe, and to get user's height and weight for the BMI calculator

Graphic OLED Display:

The display is used to display the menu and all health metrics for the user to view

Software

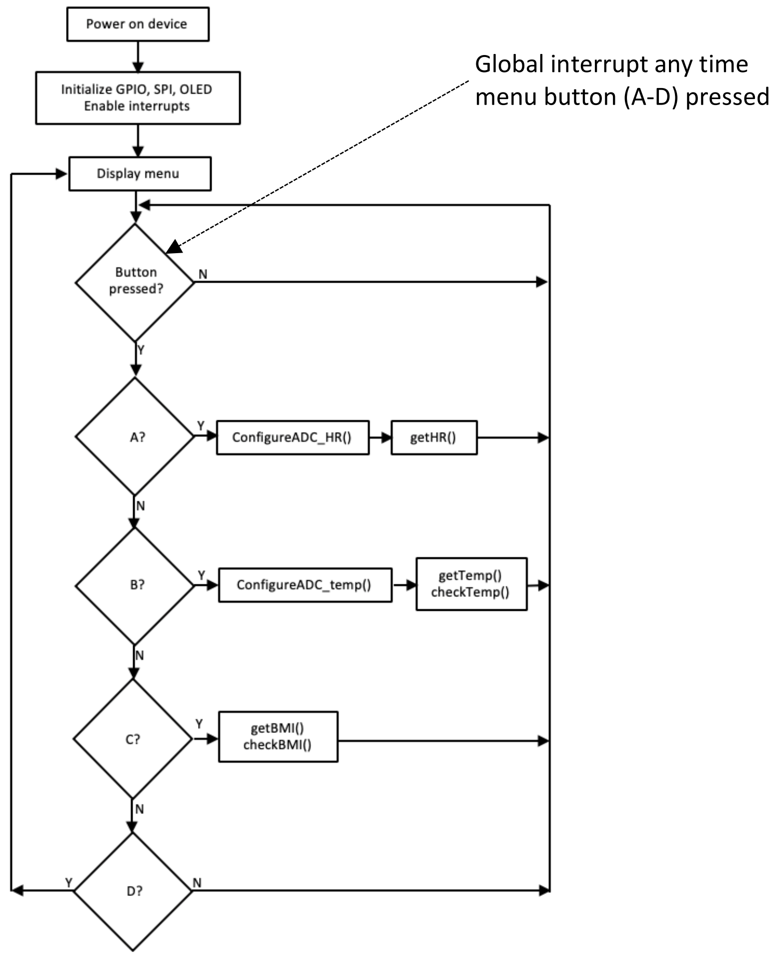

Figure 2 High-Level Overview of Software

Basic Functionality

When power is first delivered to the device/microcontroller GPIOs, OLED, and SPI are configured, and global interrupts are turned on. The menu appears on the display and waits for the user to select the health metric they wish to view. When a button is pressed, the program checks if the button pressed was A, B, C, or D and performs the specified functions for the pressed button. If the button pressed wasn't A, B, C, or D the program remains on the menu display. If at any time throughout the program buttons A, B, C, or D are pressed, the program moves to the interrupt service routine and switches to the user specified health metric.

Keypad

To determine which button is pressed on the keypad, all columns are set as inputs and all rows are set as outputs. Both the rows are column are set to a default state of '1'. The program determines which button was pressed by setting a single row low. If a button on that row is pressed, one of the four columns will also be low.

Graphic OLED Display

The display is written to using instructions and commands specified in its datasheet. The controller writes 4 bits at a time to the screen to a location on the screen specified in the program. Due to the controller always writing four bits at a time, issues were encountered when displaying the ECG graph because one ADC value was written to 4 pixels. We were able to assign one ADC value to one pixel through additional program code that took four ADC values and compared their values to determine how many pixels the value received. This allowed us to show more periods of the ECG waveform on the display and helped to smooth out the waveform. The display we chose did not come with a font library, so we had to pixel map the letters, numbers, and special characters ourselves.

Detailed software flow diagrams for all hardware and functions are available in the appendices section.

Teamwork Division

This project will be a combination of individual work and teamwork. The majority of the hardware will be done by Raymond and the software will be done by Madison. The below table shows each member's responsibilities.

	Madison	Raymond
Hardware	Battery recharge circuitry	Amplification circuitry
	Thermometer circuitry	Filtering circuitry
		Power board circuitry
Software	ADC - thermometer	OLED code
	Heart rate calculation	ADC – ECG graph
	BMI calculation	
	Keypad code	
PCB Design	Battery recharge circuitry	Amplification and filtering circuitry
	Microprocessor breakout board	Power board circuitry
	Thermometer circuitry	

Table 3 Teamwork Division

Project Schedule

I. ~2 weeks spent researching for project
II. ~5 weeks spent on hardware design and testing and PCB design
III. ~5 weeks spent on software and interfacing with hardware
IV. ~2 weeks spent on device assembly and debugging

ACTIVITY	ACTUAL START	ACTUAL DURATION
Introduction (team)	1	3
Research/Project Proposal (team)	1	2
Amplifier Circuitry (Ray)	3	2
Filtering Circuitry (Ray)	5	2
ADC Code (Madison)	4	2
Conversion Code (Madison)	5	2
External LCD Circuitry (Ray)	7	1
External LCD Code (Ray)	9	1
SPI Code (Madison)	7	1
Battery Recharge Circuitry (Madison)	7	1
PCB Design (team)	9	1
Build & Testing (team)	12	2
Demo (team)	14	1

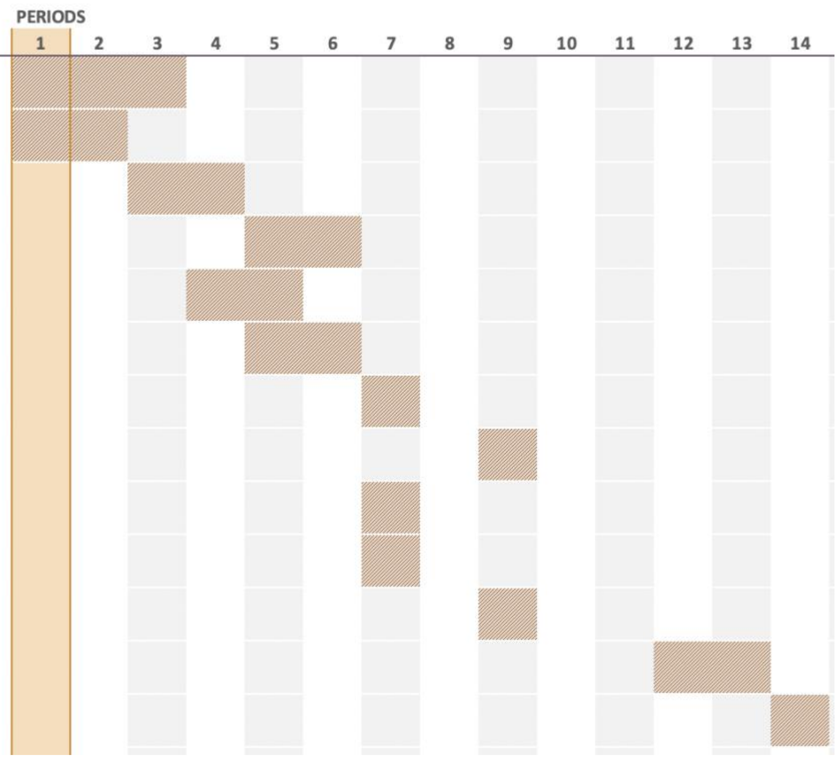

Figure 3 Gantt Chart

User Manual

1. Supply power to device → green LED turns on
2. Menu will appear on display
    A. ECG GRAPH AND HEARTRATE
    B. THERMOMETER
    C. BMI CALCULATOR
    D. RETURN TO MENU
3. Use keypad to select desired health metric
   Note: at any time during use and on any screen, pressing buttons A-D will result in switching to that health metric

⇒ If "ECG GRAPH AND HEARTRATE" selected (A):

1. Attach an electrode to left and right wrist, as close to start of palm as possible
2. Attach reference electrode
   Note: typically, on elbow or two inches below and to the right of bellybutton

3. ECG graph and heartrate (in BPM) will display on screen
    - Speaker will emit sound when heartbeat detected (peak in graph)

   Note: for best results, sit still with arms relaxed

⇒ If "THERMOMETER" selected (B):

1. Place finger around temperature sensor
2. Body temperature and range will display on screen

⇒ If "BMI CALCULATOR" selected (C):

1. Enter height in inches using keypad and press pound (#) when done
   Note: program will not let you enter more than 2 digits for height and will not move onto weight until pound is pressed
2. Enter weight in pounds using keypad and press pound (#) when done
   Note: program will not let you enter more than 3 digits for weight and will not move on to calculate BMI until pound is pressed
3. BMI and range will be displayed on screen

⇒ If "RETURN TO MENU" selected (D):

1. Program will display menu

Appendices

Software Block Diagrams

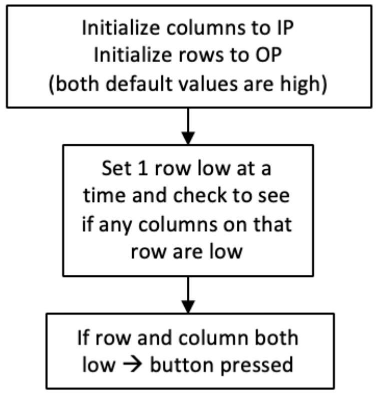

Figure 4 Keypad Flow Diagram

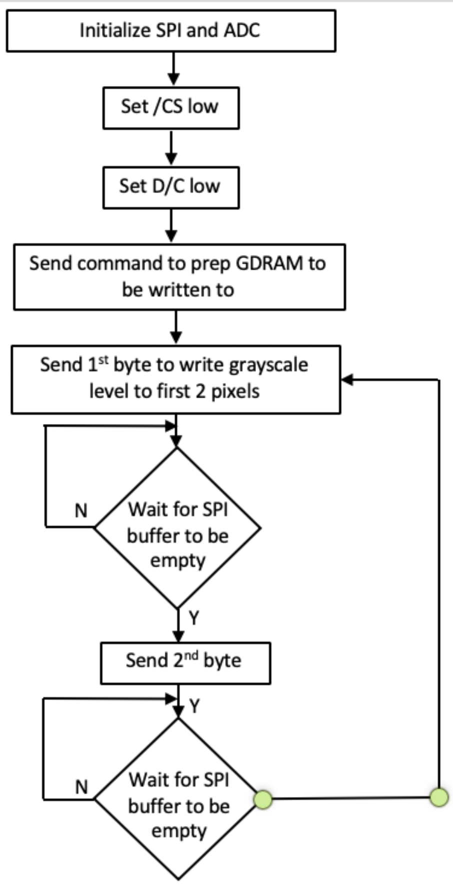

Figure 5 OLED Flow Diagram

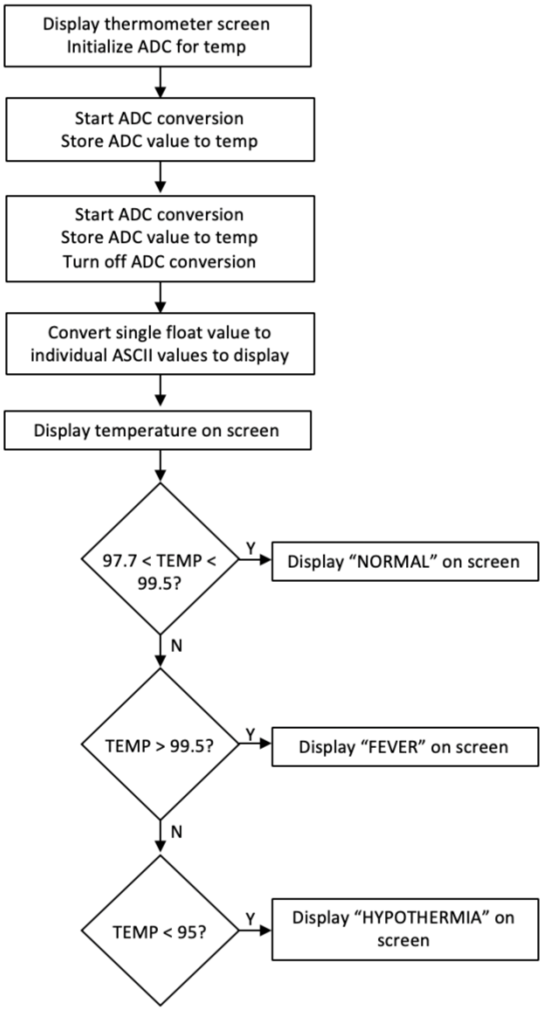

Figure 6 Thermometer Flow Diagram

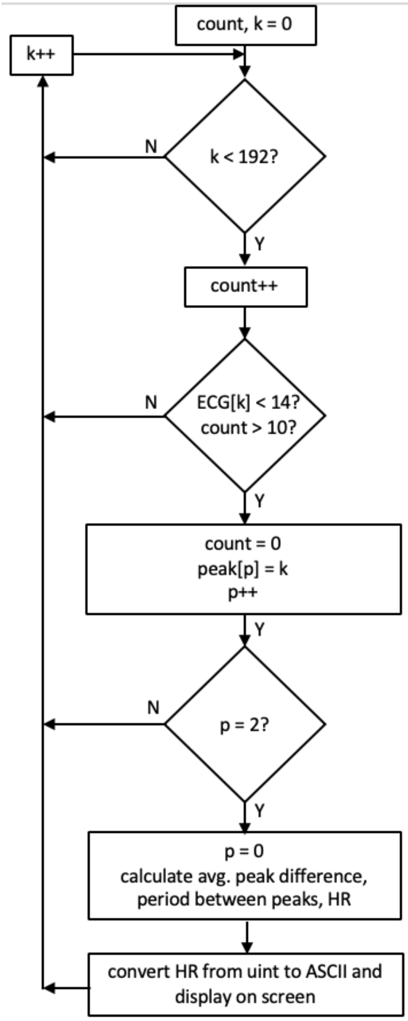

Figure 7 Heartrate Calculation Flow Diagram

Overview of Software Functions

MAIN

//Calls functions to configure + initialize GPIO pins, SPI, OLED

```
__interrupt void Port_1(void)                    //Interrupts anytime keys A-D pressed
```

CONFIGURATIONS

//Global variables for program defined in this header file

```
void configureGPIO(void);                        //Configs all GPIO pins used in program
void ConfigureADC_temp(void);                    //Configs ADC A1 to read temperature
void ConfigureADC_HR(void);                      //Configs ADC A2 to read electrode signal

uint16_t ASCIItoUint(uint16_t ASCII);            //Converts ASCII value to uint value
uint8_t UintToASCII(uint8_t uint);               //Converts uint value to ASCII value
```

FONTS

//Letters, numbers, and special characters needed for graphic display

```
void ZERO(uint8_t, uint8_t);
void ONE(uint8_t,uint8_t);
void TWO(uint8_t,uint8_t);
void THREE(uint8_t,uint8_t);
void FOUR(uint8_t,uint8_t);
void FIVE(uint8_t,uint8_t);
void SIX(uint8_t,uint8_t);
void SEVEN(uint8_t,uint8_t);
void EIGHT(uint8_t,uint8_t);
void NINE(uint8_t,uint8_t);
void POUND(uint8_t,uint8_t);
void COLON(uint8_t, uint8_t);
void PERIOD(uint8_t,uint8_t);
void SLASH(uint8_t,uint8_t);
void A(uint8_t,uint8_t);
void B(uint8_t,uint8_t);
void C_(uint8_t,uint8_t);
void D(uint8_t,uint8_t);
void E(uint8_t,uint8_t);
void F(uint8_t,uint8_t);
void G(uint8_t,uint8_t);
void H(uint8_t,uint8_t);
void I(uint8_t,uint8_t);
void J(uint8_t,uint8_t);
void K(uint8_t,uint8_t);
```

```
void L(uint8_t,uint8_t);
void M(uint8_t,uint8_t);
void N_(uint8_t,uint8_t);
void O(uint8_t,uint8_t);
void P(uint8_t,uint8_t);
void Q(uint8_t,uint8_t);
void R(uint8_t,uint8_t);
void S(uint8_t,uint8_t);
void T(uint8_t,uint8_t);
void U(uint8_t,uint8_t);
void V_(uint8_t,uint8_t);
void W(uint8_t,uint8_t);
void Y(uint8_t,uint8_t);
void X_(uint8_t,uint8_t);
void Z_(uint8_t, uint8_t);
void AST(uint8_t,uint8_t);
```

KEYPAD

```
uint8_t getKey_MENU(void);                      //Obtains health metric selection: keys A-D
uint8_t getKey_BMI(void);                       //Obtains height +  weight: all keys
```

OLED

```
//Functions to use display
void spi_init();                                //Initializes SPI for OLED use
void oled_init();                               //Initializes OLED according to datasheet
void SetCommandLock(uint8_t, uint8_t);          //Basic commands for OLED
void SleepMode(uint8_t);                        //Turns display on and off
void ClockSet(uint8_t, uint8_t);                //Sets clock divider
void MultiplexRatio(uint8_t, uint8_t);          //Set MUX ratio
void DisplayOffset(uint8_t, uint8_t);           //Sets vertical shift
void DisplayStartLine(uint8_t, uint8_t);        //Set display RAM display start line
void Remap(uint8_t, uint8_t);                   //Determines how data appears on display
void VDDRegulator(uint8_t, uint8_t);            //Selects either external or internal VDD
void Contrast(uint8_t, uint8_t);                //Sets 1 out of 256 contrast steps
void MasterContrast(uint8_t, uint8_t);          //Sets contrast for entire screen
void LinearGrayscale(uint8_t);                  //Sets linear grayscale table
void PhaseLength(uint8_t, uint8_t);             //Sets pulse width for linear grayscale table
void PrechargeVoltage(uint8_t, uint8_t);        //Sets pre-charge voltage level
void PrechargePeriod(uint8_t, uint8_t);         //Sets VCOMH voltage level
void VCOMH(uint8_t, uint8_t);                   //Sets VCOMH voltage level
void DisplayMode(uint8_t);                      //Sets entire display mode
void ExitPartialDisplay(uint8_t);               //Turns off partial display mode
void SendRAMData(uint8_t);                      //Data to be written to RAM
void SetColumnAddress(uint8_t, uint8_t, uint8_t); //Sets column start and end address
```

```
void SetRowAddress(uint8_t, uint8_t, uint8_t);          //Sets row start and end address
void PrepRAM(uint8_t);                                  // Enables MCU to write data into RAM
void SetGPIO(uint8_t, uint8_t);                         //GPIO pins used for OLED
void ClearRAM(void);                                    //Clears entire display
void ClearRAM_ECG (void);                               //Clears ECG graph
void erase(uint8_t,uint8_t,uint8_t);                    //Erases data on display
void delay_ms(unsigned int ms);                         //Delay clock cycles function

//Display Screens
void FirstScreen(void);                                 //MENU
void ThermometerScreen(void);                           //THERMOMETER
void BMIScreen(void);                                   //BMI
void HeartrateScreen(void);                             //ECG + HR
uint8_t valueToDisplay(uint8_t value, uint8_t colstart, uint8_t rowstart);   //Displays results

//Results for thermometer and BMI
void TempResults(uint8_t result);
void BMIResults(uint8_t result);
```

HEALTH DATA

```
//Heart Rate Functions
void getHR(void);                                       //Reads ADC + displays ECG, calculates HR

//BMI Functions
void getBMI(void);                                      //Calculates + displays BMI
void checkBMI(uint8_t BMI);                             //Checks BMI range and displays on screen

//Thermometer Functions
void getTemp(void);                                     //Reads ADC + displays body temp
void checkTemp(void);                                   //Checks temp range and displays on screen
```

Schematics

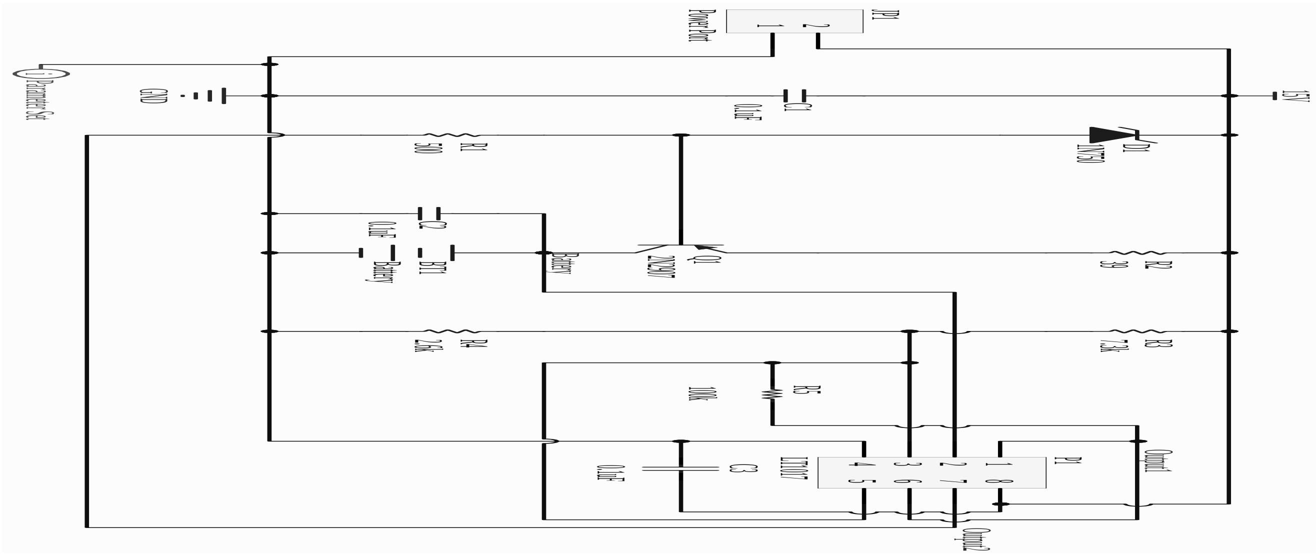

Figure 9 Battery Recharge Circuit Schematic

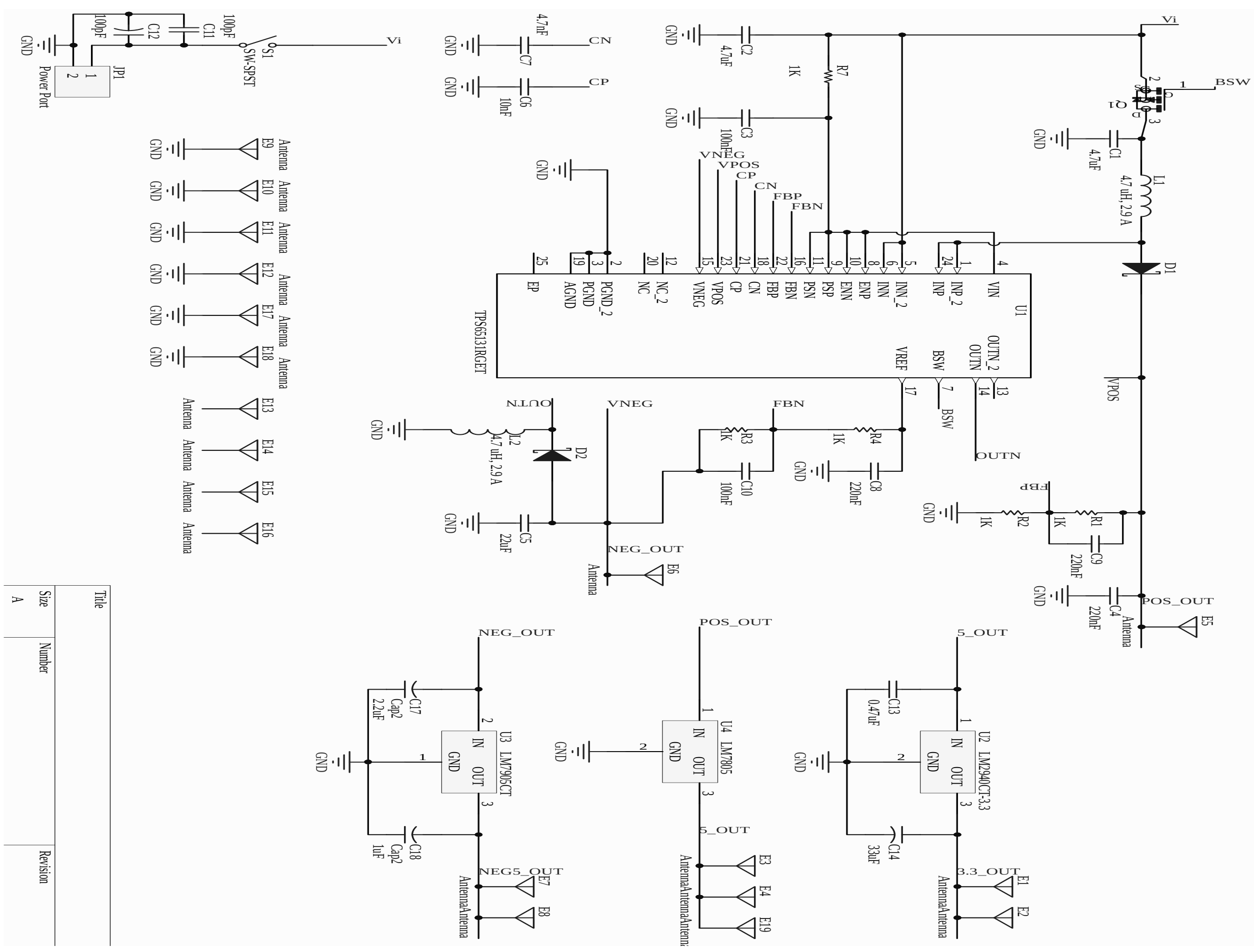

Figure 8 Power Supply Circuit Schematic

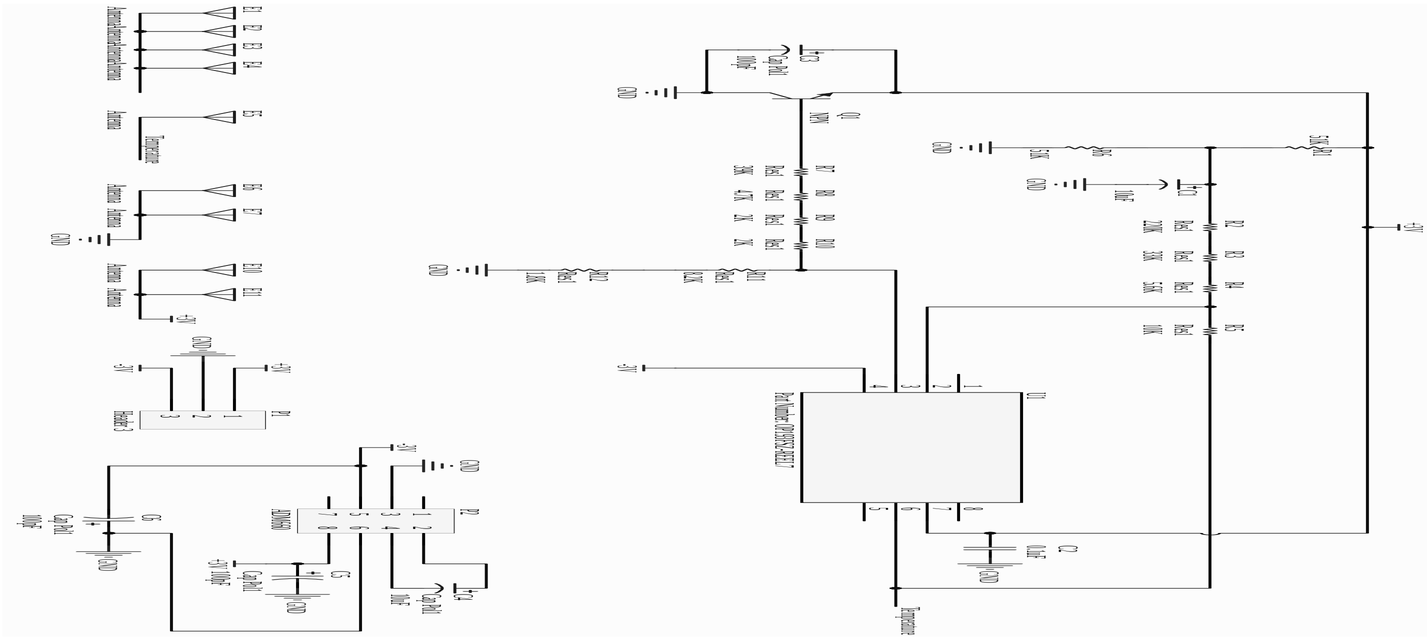

Figure 10 Thermometer Circuit Schematic

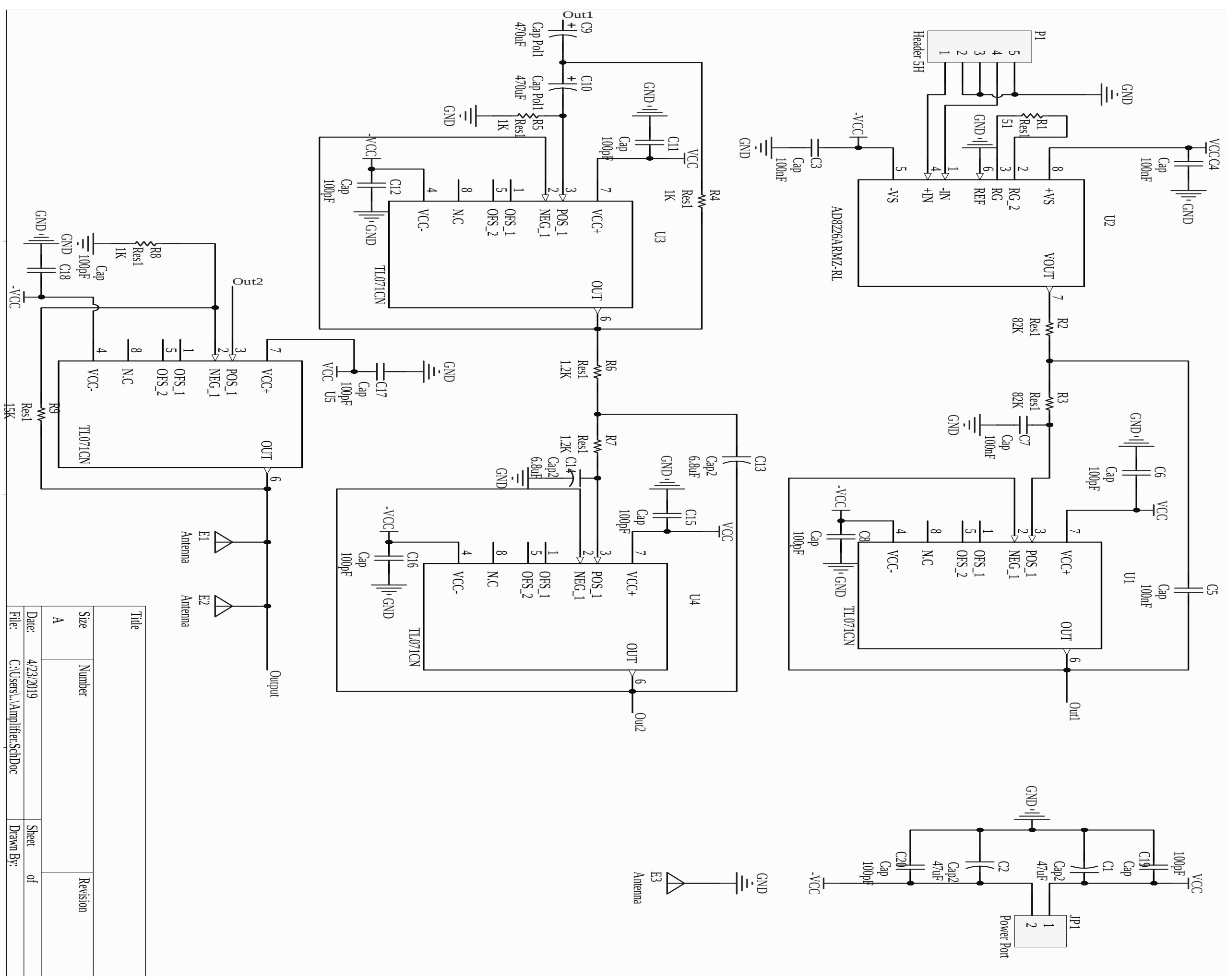

Figure 11 Amplifier and Filtering Circuit Schematic

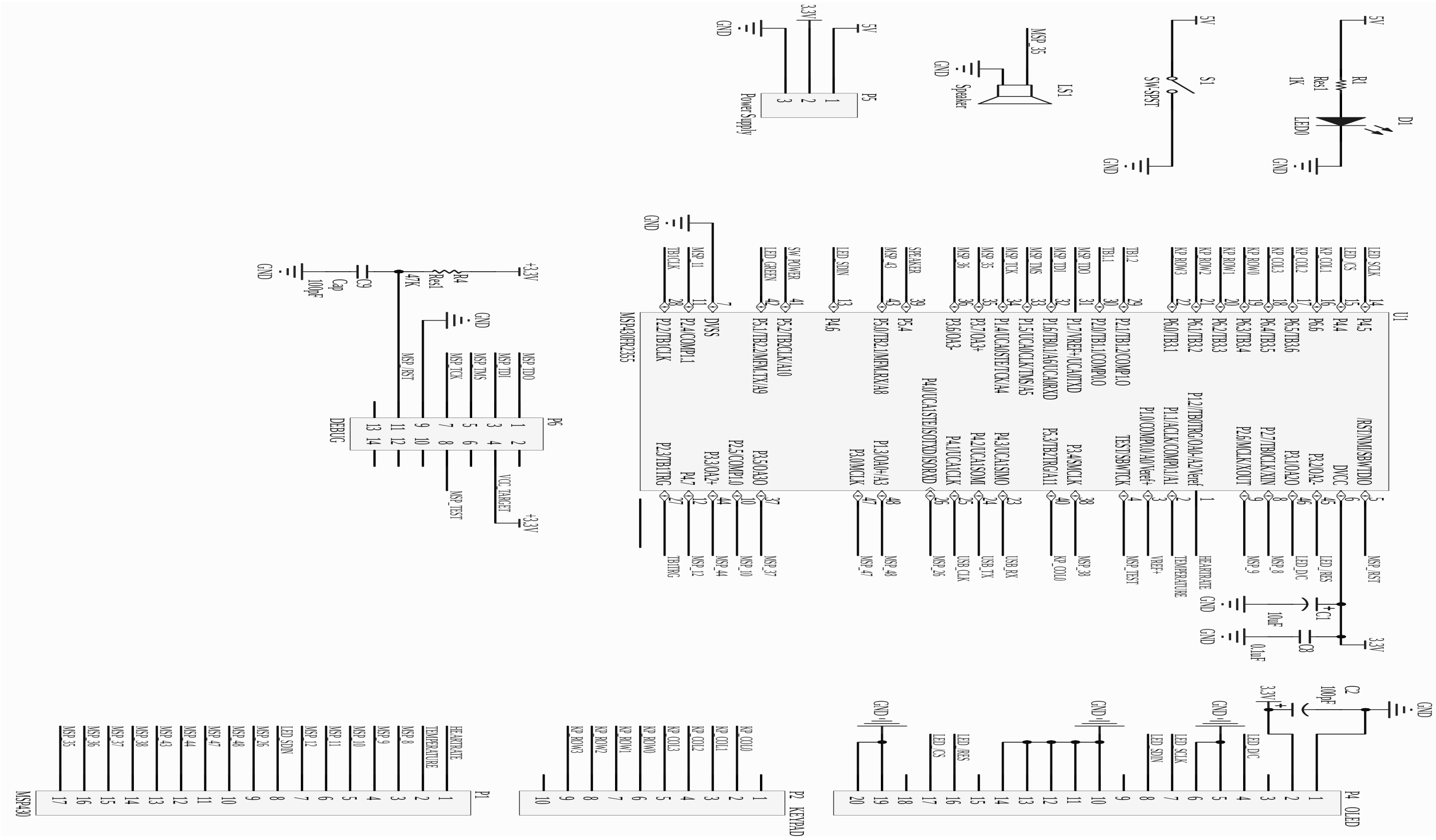

Figure 12 Main Board Schematic

Waveforms

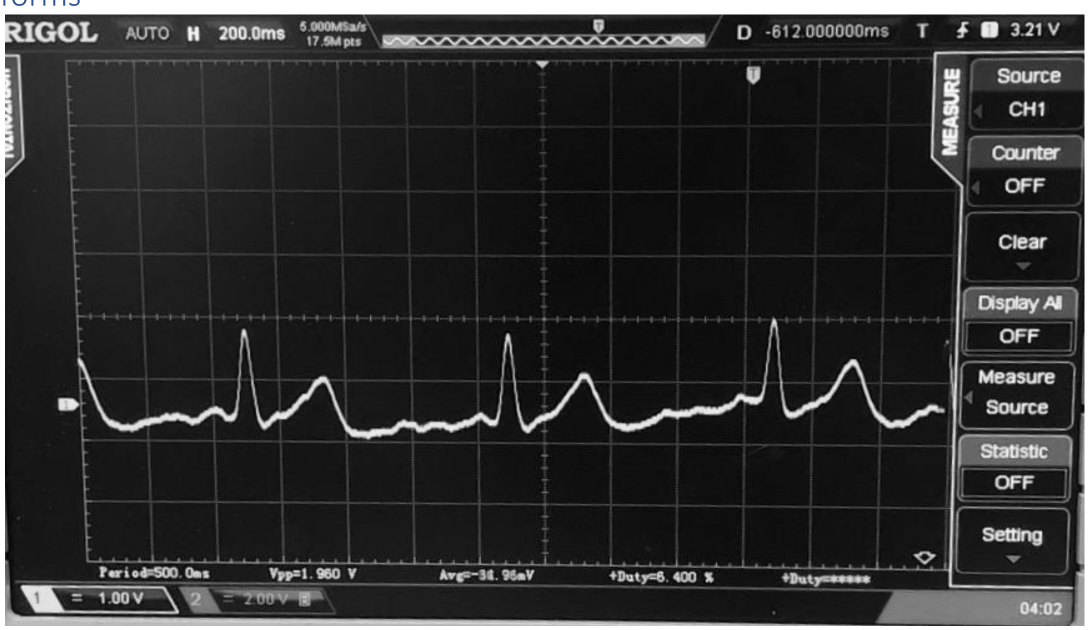

Figure 13 ECG Waveform